



# UNITED STATES PATENT AND TRADEMARK OFFICE

*my*

UNITED STATES DEPARTMENT OF COMMERCE  
United States Patent and Trademark Office  
Address: COMMISSIONER FOR PATENTS  
P.O. Box 1450  
Alexandria, Virginia 22313-1450  
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/814,546	03/31/2004	Zhibin Wang	BEAS-01514US0	7403
23910	7590	02/21/2008		
FLIESLER MEYER LLP 650 CALIFORNIA STREET 14TH FLOOR SAN FRANCISCO, CA 94108			EXAMINER BUI, HANH THI MINH	
			ART UNIT	PAPER NUMBER
			2192	
			MAIL DATE	DELIVERY MODE
			02/21/2008	PAPER

**Please find below and/or attached an Office communication concerning this application or proceeding.**

The time period for reply, if any, is set in the attached communication.

<b>Office Action Summary</b>	<b>Application No.</b> 10/814,546	<b>Applicant(s)</b> WANG, ZHIBIN	
	<b>Examiner</b> Hanh T. Bui	<b>Art Unit</b> 2192	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

#### Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

#### Status

- 1) ☒ Responsive to communication(s) filed on 31 March 2004.
- 2a) ☐ This action is **FINAL**.                      2b) ☒ This action is non-final.
- 3) ☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

#### Disposition of Claims

- 4) ☒ Claim(s) 1-21 is/are pending in the application.
- 4a) Of the above claim(s) \_\_\_\_\_ is/are withdrawn from consideration.
- 5) ☐ Claim(s) \_\_\_\_\_ is/are allowed.
- 6) ☒ Claim(s) 1-21 is/are rejected.
- 7) ☐ Claim(s) \_\_\_\_\_ is/are objected to.
- 8) ☐ Claim(s) \_\_\_\_\_ are subject to restriction and/or election requirement.

#### Application Papers

- 9) ☐ The specification is objected to by the Examiner.
- 10) ☒ The drawing(s) filed on \_\_\_\_\_ is/are: a) ☒ accepted or b) ☐ objected to by the Examiner.  
Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).  
Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).
- 11) ☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

#### Priority under 35 U.S.C. § 119

- 12) ☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).
- a) ☐ All    b) ☐ Some \* c) ☐ None of:
1. ☐ Certified copies of the priority documents have been received.
  2. ☐ Certified copies of the priority documents have been received in Application No. \_\_\_\_\_.
  3. ☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

\* See the attached detailed Office action for a list of the certified copies not received.

#### Attachment(s)

- |  |   |
|--|---|
| 1) <input checked="" type="checkbox"/> Notice of References Cited (PTO-892)                                | 4) <input type="checkbox"/> Interview Summary (PTO-413)<br>Paper No(s)/Mail Date. _____ |
| 2) <input type="checkbox"/> Notice of Draftsperson's Patent Drawing Review (PTO-948)                       | 5) <input type="checkbox"/> Notice of Informal Patent Application                       |
| 3) <input type="checkbox"/> Information Disclosure Statement(s) (PTO/SB/08)<br>Paper No(s)/Mail Date _____ | 6) <input type="checkbox"/> Other: _____  |

### **DETAILED ACTION**

1. This is response to application filed on March 31<sup>st</sup>, 2004 in which claims 1 to 21 are presented for examination.

### ***Status of Claims***

2. Claims 1 to 21 are pending, of which claims 1, 8 and 15 are in independent form.

### ***Oath/Declaration***

3. The Office acknowledges receipt of a properly signed oath/declaration filed on March 31<sup>st</sup>, 2004.

### ***Priority***

4. The priority date considered for this application is March 31<sup>st</sup>, 2004.

### ***Claim Objections***

5. Claims 1 is objected to because of the following informalities:

- The word "and" in sixth line should be removed.

Appropriate correction is required.

### ***Claim Rejections - 35 USC § 112***

6. The following is a quotation of the second paragraph of 35 U.S.C. 112:

The specification shall conclude with one or more claims particularly pointing out and distinctly claiming the subject matter which the applicant regards as his invention.

7. Claims 2, 8-21 are rejected under 35 U.S.C. 112, second paragraph, as being indefinite for failing to particularly point out and distinctly claim the subject matter which applicant regards as the invention.

a. Claims are rejected for lack of antecedent basis:

i. Claims 2, 9 and 16 recite the limitation "*the same computer or machine*" in second line. There is insufficient antecedent basis for this limitation in the claims.

ii. Claims 9 and 16 recite the limitation "*the system*" in first line. There is insufficient antecedent basis for this limitation in the claims.

iii. Claims 8 and 15 recite the limitation "*the generic interface commands*" in sixth line. There is insufficient antecedent basis for this limitation in the claims. Claims 9-14 and 16-21 are also rejected as being dependent on the rejected base claims 8 and 15, respectively.

### ***Claim Rejections - 35 USC § 101***

1. 35 U.S.C. 101 reads as follows:

Whoever invents or discovers any new and useful process, machine, manufacture, or composition of matter, or any new and useful improvement thereof, may obtain a patent therefor, subject to the conditions and requirements of this title.

2. Claims 1 and 4-7 are rejected under 35 U.S.C. 101 because the claimed invention is directed to non-statutory subject matter, specifically directed towards computer programs representing computer listings per se.

Claim 1 recites "*a system*" that has been reasonably interpreted as computer programs, software, listing per se. Computer programs claimed as computer listings per

se, i.e., the descriptions or expressions of the programs, are not physical "things." They are neither computer components nor statutory processes, as they are not "acts" being performed. Such claimed computer programs do not define any structural and functional interrelationships between the computer program and other claimed elements of a computer which permit the computer program's functionality to be realized. See Lowry, 32 F.3d at 1583-84, 32 USPQ2d at 1035.

Claims 4-7 do not overcome the deficiency as noted above; therefore they are also rejected as non-statutory subject matter.

### ***Double Patenting***

3. The nonstatutory double patenting rejection is based on a judicially created doctrine grounded in public policy (a policy reflected in the statute) so as to prevent the unjustified or improper timewise extension of the "right to exclude" granted by a patent and to prevent possible harassment by multiple assignees. A nonstatutory obviousness-type double patenting rejection is appropriate where the conflicting claims are not identical, but at least one examined application claim is not patentably distinct from the reference claim(s) because the examined application claim is either anticipated by, or would have been obvious over, the reference claim(s). See, e.g., *In re Berg*, 140 F.3d 1428, 46 USPQ2d 1226 (Fed. Cir. 1998); *In re Goodman*, 11 F.3d 1046, 29 USPQ2d 2010 (Fed. Cir. 1993); *In re Longi*, 759 F.2d 887, 225 USPQ 645 (Fed. Cir. 1985); *In re Van Ornum*, 686 F.2d 937, 214 USPQ 761 (CCPA 1982); *In re Vogel*, 422

F.2d 438, 164 USPQ 619 (CCPA 1970); and *In re Thorington*, 418 F.2d 528, 163 USPQ 644 (CCPA 1969).

A timely filed terminal disclaimer in compliance with 37 CFR 1.321(c) or 1.321(d) may be used to overcome an actual or provisional rejection based on a nonstatutory double patenting ground provided the conflicting application or patent either is shown to be commonly owned with this application, or claims an invention made as a result of activities undertaken within the scope of a joint research agreement.

Effective January 1, 1994, a registered attorney or agent of record may sign a terminal disclaimer. A terminal disclaimer signed by the assignee must fully comply with 37 CFR 3.73(b).

4. Claims 1-21 are provisionally rejected on the ground of nonstatutory obviousness-type double patenting as being unpatentable over claims 1-21 of copending Application No. 10/814,563.

This is a provisional obviousness-type double patenting rejection because the conflicting claims have not in fact been patented because of the following reasons:

a. Regarding claims 1, 8, and 15: copending application 10/814,563 recites the “interpretive engine”, “native library” claimed in the independent claims 1, 8, 15. It would have been obvious to one having ordinary skill in the art at the time of the invention was made to use editor to enter commands for the testing tool.

b. Regarding claims 2-7, 9-14, and 16-21: copending application 10/814,563 recites the language of claims 2-7, 9-14 and 16-21 to be identical to the instant claims 2-7, 9-14 and 16-21.

<b>INSTANT APPLICATION</b> <b>10/814,546</b>	<b>COPENDING APPLICATION</b> <b>10/814,563</b>
<p>Claim 1:  <b>A system that provides a generic user interface testing framework, comprising:</b></p> <ul style="list-style-type: none"> <li>- <b>an interpretive engine that receives and translates generic interface commands from a tester ;</b></li> <li>- <b>a native library for mapping the generic interface commands to native language understood by a particular test software tool;</b> <ul style="list-style-type: none"> <li>- <b>wherein the interpretive engine uses the native library to map the directives into tool-dependent codes that are then passed to the test software tool.</b></li> </ul> </li> <li>- <b>an editor that allows a tester to enter a number of test commands or directives, wherein the editor includes a rules-based wizard for assisting the tester in generating test commands and directive scripts;</b></li> </ul>	<p>Claim 1:  <b>A system that provides a generic user interface testing framework, and allows a user to test and debug graphical user Interfaces for software applications under development, comprising:</b></p> <ul style="list-style-type: none"> <li>- <b>an interpretive engine that executes on the computer, and that Includes a plurality of dynamically loaded libraries corresponding to the plurality of software test tools, wherein the interpretive engine receives the generic interface commands defined in the test case input file, loads required libraries to map the generic interface commands to corresponding tool-specific testing operations, invokes the software test tools to perform the testing operations on the software application's graphical user interface, and reports the success or failure of the testing operations.</b> <ul style="list-style-type: none"> <li>- <b>test case input file stored on the computer readable medium, that contains a plurality of generic interface commands, wherein the test case input file can be edited and reused as necessary by a user to specify different generic interface commands for testing against a software application's graphical user interface in the same or a different software test tool;</b></li> <li>- <b>a computer including a computer readable medium, and a processor operating thereon;</b></li> <li>- <b>a software application source code, stored on the computer readable medium, wherein the software application source code defines a software application under development, including a graphical user</b></li> </ul> </li> </ul>

	<p>interface as part of the software application, and wherein the software application source code executes on the computer to display its graphical user interface;</p> <ul style="list-style-type: none"><li>- one or more software test tools that are invoked to perform testing operations on the graphical user interface that is displayed while the software application is running</li></ul>
<p>Claim 8: <b>A method for providing a generic user interface testing framework, comprising the steps of:</b></p> <ul style="list-style-type: none"><li>- <b>allowing a tester to enter a number of generic test commands or directives via an editor or interface wherein the editor includes a rules-based wizard for assisting the tester in generating test commands and directive scripts;</b></li><li>- <b>translating, using an interpretive engine, the generic interface commands received from the tester, and mapping, using a native library, the generic commands to native language understood by a particular test software tool;</b></li><li>- <b>wherein the interpretive engine uses the native library to map the directives into tool-dependent codes that are then passed to the test software tool.</b></li></ul>	<p>Claim 8: <b>A method for providing a generic user interface testing framework that allows a user to test and debug graphical user interfaces for software applications under development, comprising the steps of:</b></p> <ul style="list-style-type: none"><li>- <b>allowing a user to enter a test case input file stored on the computer readable medium, that contains a plurality of generic interface commands, wherein the test case input file can be edited and reused as necessary by a user to specify different generic interface commands for testing against a software application's graphical user interface in the same or a different software test tool;</b></li><li>- <b>using a plurality of dynamically loaded libraries corresponding to the plurality of software test tools to receive the generic interface commands defined in the test case input file, load required libraries to map the generic interface command, to corresponding tool-specific testing operations, invoke the software test tools to perform the testing operations on the software application's graphical user interface, and report the success or failure of the test</b></li><li>- <b>executing a software application source code, wherein the software application source code defines a software application under development, including a graphical user interface as part of the</b></li></ul>

	<p>software application, and wherein the software the software application source code executes to display its graphical user interface;</p> <ul style="list-style-type: none"><li>- providing one or more software test tools that are invoked to perform testing operations on the graphical user interface that is displayed while the software application is running;</li></ul>
<p>Claim 15: <i>A computer readable medium including instructions stored thereon which when executed cause the computer to perform the steps of:</i></p> <ul style="list-style-type: none"><li>- <i>allowing a tester to enter a number of generic test commands or directives via an editor or interface wherein the editor includes a rules-based wizard for assisting the tester in generating test commands and directive scripts;</i></li><li>- <i>translating, using an interpretive engine, the generic interface commands received from the tester, and mapping, using a native library, the generic commands to native language understood by a particular test software tool;</i></li><li>- <i>wherein the interpretive engine uses the native library to map the directives into tool-dependent codes that are then passed to the test software tool.</i></li></ul>	<p>Claim 15: <b>A computer readable medium including instructions stored thereon which when executed cause the computer to perform the steps of:</b></p> <ul style="list-style-type: none"><li>- <b>allowing a user to enter a test case input file stored on the computer readable medium, that contains a plurality of generic interface commands, wherein the test case input file can be edited and reused as necessary by a user to specify different generic interface commands for testing against a software applications graphical user interface in the same or a different software test tool;</b></li><li>- <b>using a plurality of dynamically loaded libraries corresponding to the plurality of software test tools to receive the generic interface commands defined in the test case input file, load required libraries to map the generic interface commands to corresponding tool-specific testing operations, invoke the software test tools to perform the testing operations on the software application's graphical user interface, and report the success or failure of the testing operations.</b></li><li>- <b>executing a software application source code, wherein the software application source code defines a software application under development, including a</b></li></ul>

	<i>graphical user interface as part of the software application, and wherein the software the software application source code executes to display its ,graphical user interface; - providing one or more software test tools that are invoked to perform testing operations on the graphical user interface that is displayed while the software application is running;</i>
--	---

2. Claims 1-21 are provisionally rejected on the ground of nonstatutory obviousness-type double patenting as being unpatentable over claims 1-21 of copending Application No. 10/814,200 in view of McNeely et al. (Pub. No. 2002/0162059 – hereinafter, McNeely).

This is a provisional obviousness-type double patenting rejection because the conflicting claims have not in fact been patented because of the following reasons:

a. Regarding claims 1, 8, and 15: copending application 10/814,200 recites the “interpretive engine”, “native library” claimed in the independent claims 1, 8, 15. It would have been obvious to one having ordinary skill in the art at the time of the invention was made to use editor to enter commands for the testing tool.

b. Regarding claims 2-7, 9-14, and 16-21: copending application 10/814,200 recites the language of claims 2-7, 9-14 and 16-21 to be identical to the instant claims 2-7, 9-14 and 16-21.

<b>INSTANT APPLICATION 10/814,546</b>	<b>COPENING APPLICATION 10/815,200</b>
Claim 1: <b><i>A system that provides a generic user</i></b>	Claim 1: <b><i>A system that provides a generic user</i></b>

<p><b>interface testing framework, comprising:</b></p> <ul style="list-style-type: none"> <li>- <b>an interpretive engine that receives and translates generic interface commands from a tester ;</b></li> <li>- <b>a native library for mapping the generic interface commands to native language understood by a particular test software tool;</b></li> <li>- <b>an editor that allows a tester to enter a number of test commands or directives, wherein the editor includes a <i>rules-based wizard for assisting the tester in generating test commands and directive scripts</i>;</b></li> </ul> <p>- <b>wherein the interpretive engine uses the native library to map the directives into tool-dependent codes that are then passed to the test software tool.</b></p>	<p><b>interface testing framework, comprising:</b></p> <ul style="list-style-type: none"> <li>- <b>an interpretive engine that receives and translates generic interface commands from a tester ;</b></li> <li>- <b>a native library for mapping the generic interface commands to native language understood by a particular test software tool;</b></li> <li>- <b>McNeely discloses "Test plan and test case editor 314 allows a user to create and edit individual test cases ..." (Figure 4 and the associated text, e.g., Pg. 5: paragraph [0046]). McNeely further discloses "The test cases are independent of the number of types ... test operator responsible for writing a test script need not know the device-specific commands ... (<i>rules-based wizard</i>)" – emphasis added - (See Pg. 2: lines 2-9)</b></li> <li>- <b>wherein the interpretive engine uses the native library to map the directives into tool-dependent codes that are then passed to the test software tool.</b></li> </ul>
<p>Claim 8: A method for providing a generic user interface testing framework, comprising the steps of:</p> <ul style="list-style-type: none"> <li>- <b>allowing a tester to enter a number of generic test commands or directives via an editor or interface wherein the editor includes a <i>rules-based wizard for assisting the tester in generating test commands and directive scripts</i>;</b></li> </ul> <p>- <b>translating, using an interpretive engine, the generic interface</b></p>	<p>Claim 8: A method for providing a generic user interface testing framework, comprising the steps of:</p> <ul style="list-style-type: none"> <li>- <b>allowing a tester to enter a number of generic test commands or directives via an editor or interface</b></li> </ul> <p>McNeely discloses "Test plan and test case editor 314 allows a user to create and edit individual test cases ..." (Figure 4 and the associated text, e.g., Pg. 5: paragraph [0046]). McNeely further discloses "The test cases are independent of the number of types ... test operator responsible for writing a test script need not know the device-specific commands ... (<i>rules-based wizard</i>)" – emphasis added - (See Pg. 2: lines 2-9)</p> <p>- <b>translating, using an interpretive</b></p>

<p><b>commands received from the tester, and mapping, using a native library, the generic commands to native language understood by a particular test software tool;</b></p> <p><b>- wherein the interpretive engine uses the native library to map the directives into tool-dependent codes that are then passed to the test software tool.</b></p>	<p><b>engine, the generic interface commands received from the tester, and mapping, using a native library, the generic commands to native language understood by a particular test software tool;</b></p> <p><b>- wherein the interpretive engine uses the native library to map the directives into tool-dependent codes that are then passed to the test software tool.</b></p>
<p>Claim 15: A computer readable medium including instructions stored thereon which when executed cause the computer to perform the steps of:</p> <p><b>- allowing a tester to enter a number of generic test commands or directives via an editor or interface wherein the editor includes a rules-based wizard for assisting the tester in generating test commands and directive scripts;</b></p> <p><b>- translating, using an interpretive engine, the generic interface commands received from the tester, and mapping, using a native library, the generic commands to native language understood by a particular test software tool;</b></p> <p><b>- wherein the interpretive engine uses the native library to map the directives into tool-dependent codes that are then passed to the test software tool.</b></p>	<p>Claim 15: A method for providing a generic user interface testing framework, comprising the steps of:</p> <p><b>- allowing a tester to enter a number of generic test commands or directives via an editor or interface</b></p> <p>McNeely discloses "Test plan and test case editor 314 allows a user to create and edit individual test cases ..." (Figure 4 and the associated text, e.g., Pg. 5: paragraph [0046]). McNeely further discloses "The test cases are independent of the number of types ... test operator responsible for writing a test script need not know the device-specific commands ... (rules-based wizard) " – emphasis added - (See Pg. 2: lines 2-9)</p> <p><b>- translating, using an interpretive engine, the generic interface commands received from the tester, and mapping, using a native library, the generic commands to native language understood by a particular test software tool;</b></p> <p><b>- wherein the interpretive engine uses the native library to map the directives into tool-dependent codes that are then passed to the test software tool.</b></p>

It would have been obvious to one having ordinary skill in the art at the time of the invention was made to combine the teachings of McNeely into the teachings of copending Application because such combination would have enhanced a test/user to be able to use an editor in an easy or friendly mode.

***Claim Rejections - 35 USC § 102***

2. The following is a quotation of the appropriate paragraphs of 35 U.S.C. 102 that form the basis for the rejections under this section made in this Office action:

A person shall be entitled to a patent unless –

(b) the invention was patented or described in a printed publication in this or a foreign country or in public use or on sale in this country, more than one year prior to the date of application for patent in the United States.

3. Claims 1-21 are rejected under 35 U.S.C. 102(b) as being anticipated by McNeely et al. (Pub. No. 2002/0162059 – hereinafter, McNeely).

Regarding claim 1:

McNeely discloses *a system that provides a generic user interface testing framework, comprising:*

- *an interpretive engine that receives and translates generic interface commands from a tester ;*

(Figure 6, 7 and the associated text, e.g., Pg. 8: lines 4-5 and Pg. 9: paragraph [0088]; “Execution **engine** 400 reads (**receives**), interprets (**translates**), and executes **commands** in a test plan or test case file” and “The execution **engine** 400 **reads** an abstract command language **command** (ST4) and, based on the mapping provided by

the appropriate communication interface package, **interprets the command** within the context of the specific DUT to which the command refers (ST5)." - emphasis added.).

- *a native library for mapping the generic interface commands to native language understood by a particular test software tool;*

(Pg. 4: paragraph [0044], lines 3-6 and Pg.9: paragraph [0088]; "A communication interface **package** is a software entity that defines the **mapping** between device-specific command line **interface commands** and common or device-independent abstract command language commands" and "... The SG-specific communication interface **package** that was obtained from CIP library 356 **contains** the abstract-command-language-to-tool-command-language mapping information ..." - emphasis added.).

- *an editor that allows a tester to enter a number of test commands or directives, wherein the editor includes a rules-based wizard for assisting the tester in generating test commands and directive scripts;*

(Figure 4 and the associated text, e.g., Pg. 5: paragraph [0046], Pg. 7 lines 3-6, Pg. 2: lines 2-9; "Test plan and test case **editor** 314 allows a user to **create and edit** individual **test cases** ...", "**Editor** 314 is also adapted to interface with the **graphical user interface** 310 ..." and "The test cases are independent of the number of types ... test operator responsible for writing a test script **need not know the device-specific commands** ... (**rules-based wizard**)" – emphasis added.).

- *wherein the interpretive engine uses the native library to map the directives into tool-dependent codes that are then passed to the test software tool.*

(Figures 6, 7 and the associated text, e.g., Pg. 9: paragraph [0089], lines 1-2; "the resulting **tool command** language command is subsequently **passed** to the communication interface 420" - emphasis added.).

Regarding claim 2:

McNeely discloses *the system of claim 1 wherein the system includes the test software tool stored locally on the same computer or machine.*

(Pg. 2: paragraph [0010], last three lines; "The client/server framework allows a client to be located on any system in the network, even on **the same system** on the which the server resides" - emphasis added.).

Regarding claim 3:

McNeely discloses *the system of claim 1 wherein the test software tool is stored at another computer or machine.*

(Figure 3 and the associated text, e.g., Pg. 3: paragraph [0037], lines 1-30 and paragraph [0039], lines 1-4; "Test management system client 214 communicates with test tools server 210 via network 216 and allows a user to perform a number of activities associated with test system" and "test management system client 214 accesses a test plan and associated test case files from test tools server 210 ...".

Examiner noted that in order to perform the testing, client has to communicate with server, therefore the test software tool is stored at another computer or machine.).

Regarding claim 4:

McNeely discloses *the system of claim 1 wherein the editor provides a graphical interface to allow the tester to enter said test commands.*

(Figure 4 and the associated text, e.g., Pg. 5: paragraph [0046] and Pg. 7 lines 3-6; "Test plan and test case **editor** 314 allows a user to **create and edit** individual **test cases ...**" and "**Editor** 314 is also adapted to interface with the **graphical user interface** 310 ..." - emphasis added.).

Regarding claim 5:

McNeely discloses *the system of claim 1 wherein the editor communicates the test commands as a script of directives.*

(Pg. 2: paragraph [0012], line 5; "test case may be a file of **commands** or **directives**" - emphasis added.).

Regarding claim 6:

McNeely discloses *the system of claim 1 wherein the test commands can be created offline and subsequently communicated to the interpretive engine.*

(Pg. 9: line 2; "downloads the test to execution engine 4000". Examiner noted that the test case was created offline and then downloaded to the engine via network communication.).

Regarding claim 7:

McNeely discloses *the system of claim 1 wherein the test software tool can be removed and replaced with another test software tool.*

(Pg. 7: paragraph [0065], last line; “a suitable GUI tester is added via a new package”. Examiner noted that when there are unsuitable test software tools, then that test can be removed and replaced with new test software tools.).

Regarding claim 8:

This is another method version of the rejected claim 1 above, wherein all the limitations of this claim have been noted in the rejection of claim 1.

Regarding claim 9:

The rejection of base claim 8 is incorporated. All the limitations of this claim have been noted in the rejection of claim 2.

Regarding claim 10:

The rejection of base claim 8 is incorporated. All the limitations of this claim have been noted in the rejection of claim 3.

Regarding claim 11:

The rejection of base claim 8 is incorporated. All the limitations of this claim have been noted in the rejection of claim 4.

Regarding claim 12:

The rejection of base claim 8 is incorporated. All the limitations of this claim have been noted in the rejection of claim 5.

Regarding claim 13:

The rejection of base claim 8 is incorporated. All the limitations of this claim have been noted in the rejection of claim 6.

Regarding claim 14:

The rejection of base claim 8 is incorporated. All the limitations of this claim have been noted in the rejection of claim 7.

Regarding claim 15:

This is another computer readable medium version of the rejected claim 1 above, wherein all the limitations of this claim have been noted in the rejection of claim 1.

Regarding claim 16:

The rejection of base claim 15 is incorporated. All the limitations of this claim have been noted in the rejection of claim 2.

Regarding claim 17:

The rejection of base claim 15 is incorporated. All the limitations of this claim have been noted in the rejection of claim 3.

Regarding claim 18:

The rejection of base claim 15 is incorporated. All the limitations of this claim have been noted in the rejection of claim 4.

Regarding claim 19:

The rejection of base claim 15 is incorporated. All the limitations of this claim have been noted in the rejection of claim 5.

Regarding claim 20:

The rejection of base claim 15 is incorporated. All the limitations of this claim have been noted in the rejection of claim 6.

Regarding claim 21:

The rejection of base claim 15 is incorporated. All the limitations of this claim have been noted in the rejection of claim 7.

### ***Conclusion***

8. The prior art made of record and not relied upon is considered pertinent to applicant's disclosure.

9. Any inquiry concerning this communication or earlier communications from the examiner should be directed to Hanh T. Bui whose telephone number is (571) 270-1976. The examiner can normally be reached on Mon. - Thur., 7:00AM - 3:30PM.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Tuan Dam can be reached on (571) 272-3695. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see <http://pair-direct.uspto.gov>. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free). If you would like assistance from a USPTO Customer Service Representative or access to the automated information system, call 800-786-9199 (IN USA OR CANADA) or 571-272-1000.

\_BH\_



**TUAN DAM**  
**SUPERVISORY PATENT EXAMINER**